ADL HW1 REPORT
intent classification, slot tagging

李彥霖 M11015094

Q1: Data processing (2%)
Intent classification

For each data sample in “train.json” , “eval.json” , “test.json”

1. **Origin data**:
   Text e.g. "i need you to book me a flight from ft lauderdale to houston on southwest"
   Intent:"book_flight"

2. **Tokenize**: mapping each word to a token number.
   tokens = tokenize(text) e.g. [30,99,57,74,5139,166,1028,57,24,1056,44,2674,2243,5000,2027]
   label: tokenize(label) e.g. 30

3. **Padding tokens**: padding
   padded_token = padding_tokens(tokens , max_len)
   e.g.[30,99,57,74,5139,166,1028,0,0,.....0] , here length of padding_token is 128

Then feed the padding tokens to training model

Q1: Data processing (2%)
Slot tagging

For each data sample in “train.json” , “eval.json” , “test.json”

1. **Origin data**:
   Only select the most common 3000 word, the others are set to “unknown”.
   tokens e.g. [i, want, the, west, central, neighborhood]
   tags e.g. [O, O, B-people, I-people, O, B-date, O], total has 9 tags.

2. **Tokenize**: mapping each word to a token number.
   tokens = tokenize(text) e.g. [30,99,57,74,5139,166,1028,57,24,1056,44,2674,2243,5000,2027]
   label: tokenize(label) e.g. 30

3. **Padding tokens**: padding
   padded_token = padding_tokens(tokens , max_len)
   e.g.[30,99,57,74,5139,166,1028,0,0,.....0] , length of padding_token: 128

Then feed the padding tokens to training model

Q1: Data processing (2%)
embedding

The pre-trained embedding you used.

Using glove (840B tokens, 2.2M vocab, cased, 300d vectors)

Q2: Describe your intent classification model. (2%)

1. Given input sequence $X = \{x_0, x_1, x_2, \ldots x_t\}$
2. $out, h_n, c_n = LSTM(x_t)$ Where h_n is the last step of the hidden output.
   LSTM is 2 layer bidirectional, 300 dimensional input and 512 dimension output.0.2 dropout.
3. $out = Linear(h_n)$,where Linear layer map the 512 dimension input to 150 dimension output
4. Finally , feed out to SiLU() activation function.

```
SeqClassifier(
  (embed): Embedding(6491, 300)
  (rnn): LSTM(300, 256, num_layers=2, dropout=0.2, bidirectional=True)
  (fc): Sequential(
    (0): Dropout(p=0.2, inplace=False)
    (1): Linear(in_features=512, out_features=150, bias=True)
    (2): SiLU()
  )
)
```

Q2: Describe your intent classification model. (2%)

Performance :

public score: 090488 private score: 0.91377

Loss function : Cross Entropy Loss

Optimizer: Adam

Learning rate: 1E-3

Batch size: 128

Q3: Describe your slot tagging model. (2%)

Where is the last step of the hidden output.

1. Given input sequence $X = \{x_0, x_1, x_2, \ldots x_t\}$
2. $out, h_n, c_n = LSTM(x_t)$, where out is the output of each time step. LSTM is 2 layer bidirectional, 300 dimensional input and 512 dimension output.0.2 dropout.
3. $out = Linear(out)$mapping the 512 dimension input to 256 dimension output
4. $out = Linear_2(h_n)$,mapping the 256 dimension input to 10 dimension output
5. Finally , feed out to SiLU() activation function.

Q3: Describe your slot tagging model. (2%)

Performance :

Public score: 0.81823 private score:0.82797

Loss function : Cross Entropy Loss

Optimizer: Adam

Learning rate: 1E-3

Batch size: 64

```
In [7]: model
Out[7]:
TagMultiClassifier(
  (embed): Embedding(4117, 300)
  (rnn): LSTM(300, 256, num_layers=2, batch_first=True, dropout=0.2, bidir
ectional=True)
  (fc): Sequential(
    (0): Linear(in_features=512, out_features=256, bias=True)
    (1): Dropout(p=0.2, inplace=False)
    (2): Linear(in_features=256, out_features=10, bias=True)
    (3): SiLU()
  )
)
```

Q4: Sequence Tagging Evaluation (2%)

```
e 28 avg_acc:98.09  loss:0.43
e 28 avg_acc:78.50  loss:1.87
              precision    recall  f1-score   support

        date       0.73      0.75      0.74       201
  first_name       0.83      0.96      0.89        89
   last_name       0.76      0.91      0.83        65
      people       0.68      0.72      0.70       224
        time       0.83      0.84      0.83       216

   micro avg       0.76      0.80      0.78       795
   macro avg       0.77      0.83      0.80       795
weighted avg       0.76      0.80      0.78       795
```

Q4: Sequence Tagging Evaluation (2%)

$$\text{Joint accuracy} = \frac{\text{correct sequence predicted}}{\text{number of all sequence predicted}}$$

$$\text{Token accuracy} = \frac{\text{correct token predicted}}{\text{number of all token predicted}}$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{TP}{\text{預測(tag,begin,end)為該 tag 的數量}}$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{TP}{\text{真實(tag,begin,end)為該 tag 的數量}}$$

$$\text{F1 score} = \frac{2}{\frac{1}{precision}+\frac{1}{recall}}$$ 為 precision 和 recall 的調和，

Support ＝ 真實(tag, begin, end)為該 tag 的數量

$$\text{Micro avg} = \frac{\text{所有 tag 的 TP}}{\text{所有 tag 的 TP+FP(或 FN)}}$$，

Macro avg 是對所有的 precision、recall 做平均，

Weighted avg 是根據 tag 數量做 weighted sum。